

아직 AI는 프로그래밍을 할 수 없다

박수빈

2022.11.11.

Abstract

현대에 들어서, AI는 인간의 것으로 여겨지던 많은 분야들에 참여하고 있다. 마찬가지로, 프로그래밍 역시 AI로 해결하려는 시도가 많아지고 있다. 그러나 프로그래밍과 같이 논리적인 추론이 필요한 분야에 있어서는 아직 AI에 한계점이 많다. 이러한 한계점이 해결되지 않는 이상, AI가 프로그래밍을 처음부터 끝까지 하는 것 보다는 사람의 통제 아래 유용한 도구로써 사용되는 것이 효율적으로 보인다.

현대에 들어서, 인간은 AI에게 수많은 패배를 경험하고 있다. 이세돌을 상대로 한 바둑 인공지능 AlphaGo의 승리를 시작으로, 문학이나 예술 등 인간만이 할 수 있는 것으로 여겨졌던 분야들에 점점 AI가 참여하고 있다. 프로그래밍 역시 예외가 아니다. GitHub의 Copilot을 필두로, AI를 이용해 실제로 동작하는 프로그램을 작성하려는 시도가 생겨나고 있다. AI의 발전으로 인해 프로그래머라는 직업이 AI로 대체될 수도 있다는 우려의 목소리가 나오고 있지만, 개인적으로는 AI의 프로그래밍 능력에 대해서 회의적인 시선을 가지고 있다. AI가 실제로 프로그래밍을 하려면 해결해야 될 난관이 너무 많다.

AI는 수많은 데이터로부터 학습하지만, 그것이 그 자체로 AI의 한계이다. 수많은 사례로부터 결론을 이끌어내는 것은 귀납 추론 혹은 경험론의 과정이다. 이 귀납 추론의 영역에서, AI는 그 어떤 인간보다도 뛰어나다. 그러나 아무리 귀납 추론을 잘 하더라도 귀납 추론 자체에는 태생적인 한계가 있다. ‘러셀의 칠면조’로 대표되는 귀납법의 결점은 물론이고, 수학이나 논리학 같이 연역적인 추론이 요구되는 분야에는 큰 효과가 없다. 이것들을 공부할 때도 마찬가지다. 수학을 잘 하려면 어떻게 해야 하는가? 그저 문제집만 많이 풀어보면 되는 것인가? 문제집에만 의존해서는 처음 보는 새로운 유형의 문제를 풀 수 없을 것이다. 귀납법의 ‘검은 백조’처럼 말이다. 그러나 연역적인 사고력을 기르고, 수학의 원리를 이해하고 응용할 수 있다면 어떤 문제든 풀 수 있다. 귀납적인 방법만으로는 해결할 수 없는, 단순한 경험의 개수보다는 한순간 떠오르는 기발한 아이디어가 더 중요한 분야가 있는 것이다.

같은 이유로, AI가 프로그래머를 완전히 대체할 수 없다고 확신한다. 프로그래밍 역시 수학과 같이 연역 추론이 필요한 분야이다. 본질적으로 논리적 사고와 아이디어 발상에 기반을 두고 있기 때문이다. AlphaGo의 개발사 DeepMind에서 개발한 AlphaCode의 사례를 보자. AlphaCode는 온라인 프로그래밍 대회 사이트 Codeforces의 문제를 해결하는 코드를 작성하는 인공지능이다. AlphaCode는 놀라울 정도로 문제가 요구하는 바를 잘 이해하고 이를 C++ 혹은 Python 코드로 작성해낸다. 그러나 직접 Codeforces에서 활동해본 개인적인 경험에 비추어 봤을 때, AlphaCode가 실제로 문제를 푼다기보다는 이를 코드로 옮기는 번역기에 가깝다는 인상을 받았다. 문제에 써진 내용을 그대로 구현하는 데에는 확실한 강점을 보였지만, 푸는 데에 복잡한 아이디어가 필요하거나 알고리즘을 최적화해야만 통과할 수 있는 문제들은 풀지 못하는 모습을 보였기 때문이다.

프로그래밍의 핵심은 요구하는 바를 그저 코드로 옮기는 데에 있는 것이 아니다. 코드를 작성하는 것은 정말로 누구나 할 수 있다. 프로그래밍 실력은 프로그램의 시간복잡도를 줄이는 아이디어를 떠올릴 수 있는 능력으로 가려지는 것이다. 그러나 현재의 AI는 아이디어 발상을 할 만한 응용력과 창의력을 갖추지 못한 것으로 보인다. 과연 AI가 그러한 능력을 갖추고 인간과 대등하게 프로그래밍을 할 수 있을 것인가? 이는 현재의 AI가 풀어야 할 난제이지만, 개인적으로는 AI가 근본적으로 귀납 추론에 기반을 두고 있기 때문에 이 문제를 해결하기는 정말 어려운 것으로 생각된다.

따라서, 현재로서는 AI가 인간 프로그래머의 조수로서 활용되는 것이 이상적으로 보인다. AI의 강점은 학습 데이터를 기반으로 최대한 ‘그럴듯한’ 것들을 만들어 내는데에 있다. 그러나 프로그래밍은 엄격하고 엄밀해야 한다. 수학이나 프로그래밍 같은 분야에서 99퍼센트 맞는 답은 그냥 오답이다. 특히나 프로그래밍을 할 때에는 프로그램의 실행 시간을 최적화하거나 보안을 고려하는 등 고도의 능력이 요구된다. 아직 AI의 강점은 고도의 작업을 하는 것 보다는 단순 노동에 있기 때문에, 각자의 장점을 살려서 적재적소에 활용하는 것이 가장 효율적인 방법으로 보인다.